
Bachelorarbeit

Herr
Maik Nobst

**Visionssystem für eine Robo-
teranwendung (KUKA)**

Mittweida, 16.01.2017

Bachelorarbeit

„Visionsystem“ für eine Roboteranwendung (KUKA)

Autor:

Herr

Maik Nobst

Studiengang:

BEng. Mechatronik

Seminargruppe:

Me12w1-B

Erstprüfer:

Prof. Dr. Ing. Klaus Müller

Zweitprüfer:

Prof. Dr. Ing. Swen Schmeißer

Einreichung:

Mittweida, 16.01.2017

Verteidigung/Bewertung:

Mittweida, 2017

Bibliografische Beschreibung:

Nobst, Maik:

„Visionsystem“ für eine Roboteranwendung (KUKA) – 2017

4 Seiten Verzeichnis, 33 Seiten Inhalt, 2 Seiten Anhänge

Mittweida, Hochschule Mittweida, Fakultät INW, Bachelorarbeit, 2017

Referat:

Anleitung zum Praktikum VisionTech mit Orientierungsfeststellung

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Abkürzungsverzeichnis	IV
1 Aufgabenstellung allgemein.....	1
1.1 <i>Technische Voraussetzungen und allgemeine Angaben zum Arbeitsplatz</i>	<i>2</i>
1.2 <i>Die Programme auf dem Steuerungs-PC</i>	<i>9</i>
1.3 <i>Die Programme und das UI auf dem Handheld der KRC.....</i>	<i>10</i>
2 Aufgabenstellung	13
2.1 <i>Ausgangssituation an der Roboterzelle</i>	<i>13</i>
2.2 <i>Ziel der Arbeit.....</i>	<i>13</i>
3 Durchführung	14
3.1 <i>Einrichten des Bildbereichs</i>	<i>14</i>
3.2 <i>Auswahl und Einrichten der Beleuchtung</i>	<i>15</i>
3.3 <i>Kalibrieren der Kamera.....</i>	<i>16</i>
3.4 <i>Trainieren des zu erkennenden Objekts in WorkVisual.....</i>	<i>22</i>
3.5 <i>Definieren der Null-Lage des Objekts</i>	<i>27</i>
3.6 <i>Einlernen (teachen) der Zeigerlage</i>	<i>27</i>
3.7 <i>Das zugehörige Beispiel-Programm</i>	<i>28</i>
4 Hinweise auf mögliche Fehlerquellen.....	32
4.1 <i>Einrichten der Arbeitsumgebung</i>	<i>32</i>
4.2 <i>Beim Einlernen des Objektes im WorkVisual.....</i>	<i>32</i>
4.3 <i>Beim Schreiben des Programmes</i>	<i>32</i>
5 Fazit.....	33
Literatur und Hilfen	V
Selbstständigkeitserklärung	VI

Abbildungsverzeichnis

Abbildung 1: Grobstruktur der Bildverarbeitung unter KUKA (neue Version)	2
Abbildung 2: Der Steuerungscomputer.....	3
Abbildung 3: Die Robotersteuerung.....	4
Abbildung 4: Der Roboterarm	5
Abbildung 5: Die Werkzeuge	6
Abbildung 6: Die Kamera.....	7
Abbildung 7: LED-Ringleuchte	8
Abbildung 8: WorkVisual-Bedienoberfläche.....	9
Abbildung 9: TeamViewer-Bedienoberfläche.....	9
Abbildung 10: Oberfläche des Touchpanels	10
Abbildung 11: Untermenüs im UI.....	11
Abbildung 12: VisionTech-Menü.....	12
Abbildung 13: Arbeitsbereich (Kasten)	14
Abbildung 14: Beleuchtung 1.....	15
Abbildung 15: Beleuchtung 2.....	15
Abbildung 16: Kalibrierplatte.....	16
Abbildung 17: Symbol VisionTech	16
Abbildung 18: Symbol Livebild	16
Abbildung 19: Livebild Kalibrierung	17
Abbildung 20: Button Kalibrierung	17
Abbildung 21: Untermenü Kalibrierkörper	17
Abbildung 22: Auswahl Kalibrierassistent	18
Abbildung 23: Fortsetzung Kalibrierassistent	18
Abbildung 24: Kalibrierungsvorgang	18
Abbildung 25: Neue Kalibrierebene	19
Abbildung 26: neue Basis einmessen.....	19
Abbildung 27: Lage des Basiskoordinatensystem	21
Abbildung 28: Button Taskkonfiguration	22
Abbildung 29: Auswahl Kamerasensor	22
Abbildung 30: VisionPro-Toolbox	23
Abbildung 31: Vervollständigen der Werkzeugliste	24
Abbildung 32: PatMax-Menü	24

Abbildungsverzeichnis	III
Abbildung 33: Parameterausführung	25
Abbildung 34: Winkeleinstellung.....	25
Abbildung 35: Ordnerstruktur für Taskkonfiguration.....	26
Abbildung 36: Sensorauswahl	26
Abbildung 37: Anfahren des Nullpunkts.....	27
Abbildung 38: Beispielprogramm Teil 1	28
Abbildung 39: Beispielprogramm Teil 2	29
Abbildung 40: Beispielprogramm Teil 3	30

Abkürzungsverzeichnis

KRL	KUKA-Roboter-Language
KRC	KUKA-Roboter-Control
TCP	Tool-Center-Point
UI	User Interface
VT	Vision-Tech

1 Aufgabenstellung allgemein

Die Aufgabenstellung des Bachelorprojektes an der Lehrzelle der Firma KUKA ist, die bereits angeschlossene und installierte Kamera so zu nutzen, dass der TCP erkannte Teile erreichen, deren Orientierung feststellen und diese aufnehmen kann.

Dazu soll ein „Demoprogramm“ zum Erkennen und Aufnehmen beliebiger Teile entwickelt und optimiert werden. Der Schwerpunkt soll hierbei sein, die Lage und Orientierung der Teile zu erkennen und richtig zu übergeben. In der Folge dessen sollen Bewegungen des Roboters daraus veranlasst werden.

Die Ausbildung an der Lehrzelle setzt zuvor erlangtes Wissen in Bezug auf die Handhabung und Programmierung voraus. In diesem Fall wird das Vorwissen zur Bedienung und Programmierung des Robotersystems benötigt. Hierzu gehören der Umgang und das Verständnis für Roboterbewegung, Beschleunigung und das Bremsverhalten eines Industrieroboters. Weitere vorausgesetzte Kenntnisse sind das Wissen über die Programmstruktur und den Aufbau eines Programmes. Der Aufbau und die Logik hinter der Programmierung des WorkVisual sollte erlangt und verinnerlicht worden sein.

1.1 Technische Voraussetzungen und allgemeine Angaben zum Arbeitsplatz

Die technischen Voraussetzungen umschließen sowohl den vollständigen Roboterarbeitsplatz mit KRC-Steuerung und einem dazugehörigen PC mit Ethernetverbindung zur Robotersteuerung (KRC-KUKA Roboter Control).

Die am Roboter vorhandene Steuerung sollte eine funktionsfähige und kamerakompatible Version von „VisionTech“ enthalten, um mit dem Kamerasystem kommunizieren zu können. Es sollten für alle benötigten Programme und Funktionen die Lizenzen erworben und eingegeben werden. Dies ist für den reibungslosen Umgang mit der Software notwendig und sollte unbedingt eingehalten werden. Die KRC an sich, wozu der Roboterarm mit Werkzeugen wie Sauger, welchen wir für das fertige Programm benötigen, und dem Greifer, welcher zum Einrichten und Kalibrieren benötigt wird. Diese Werkzeuge werden pneumatisch, also mit Luft, betrieben. Ebenso zur Roboterzelle gehört die Kamera mit angebauter LED-Ringleuchte, welche für die bestmögliche Ausleuchtung sorgt. Im Folgenden werden die einzelnen Teile und die wichtigsten Programme des Systems noch einmal einzeln vorgestellt und mit Funktionen und Funktionsweise erklärt.

Grobstruktur der Bildverarbeitung unter KUKA in der neuen Version.

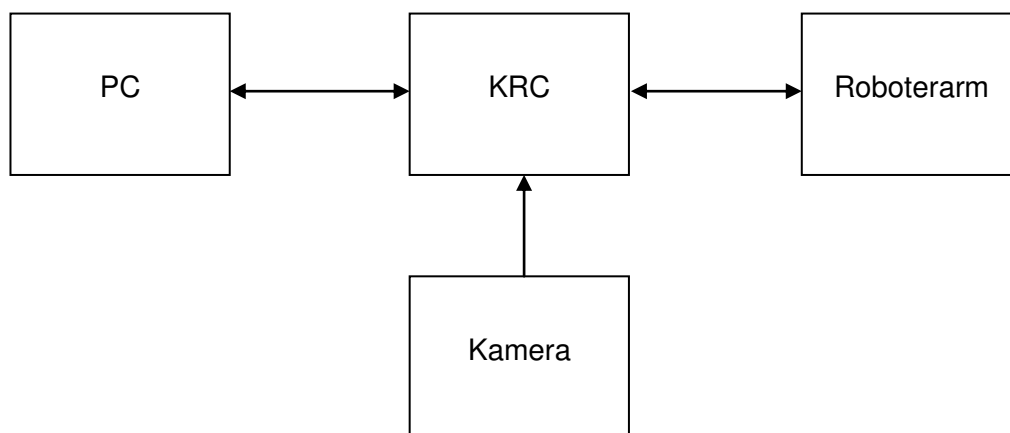


Abbildung 1: Grobstruktur der Bildverarbeitung unter KUKA (neue Version)

Die von der Kamera aufgenommenen Bilder werden über die KRC-Steuerung an den PC gesendet. Am PC werden die aufgenommenen Bilder im VisionTech in ein Objekt verarbeitet.

Dieses Objekt wird von der KRC-Steuerung in einen Bewegungsbefehl umgewandelt und an den Roboterarm abgegeben.

Im laufenden Prozess ist das Objekt bereits eingelernt und das von der Kamera aufgenommene Bild wird über das Tool im KRC direkt verarbeitet.

Der Steuerungscomputer



Abbildung 2: Der Steuerungscomputer

- CPU: Intel® Core™ i5-4590 @ 3.30GHz
- RAM: 8,00GB
- OS: Windows 7 Enterprise 64-Bit
 - Programme:
 - TeamViewer Ver. 10.0.39052
 - WorkVisual 3.1

Die Robotersteuerung (KR C4 Compact)



Abbildung 3: Die Robotersteuerung

- CPU: Intel® Celeron® G1820 @ 2.70GHz
- RAM: 2,00GB
- OS: Windows Embedded Standard 6.1.7601 SP1
- Programme:
 - TeamViewer Ver. 10.0.39052
 - KR C-OS V8.3.12
 - VisionTech

Der Roboterarm

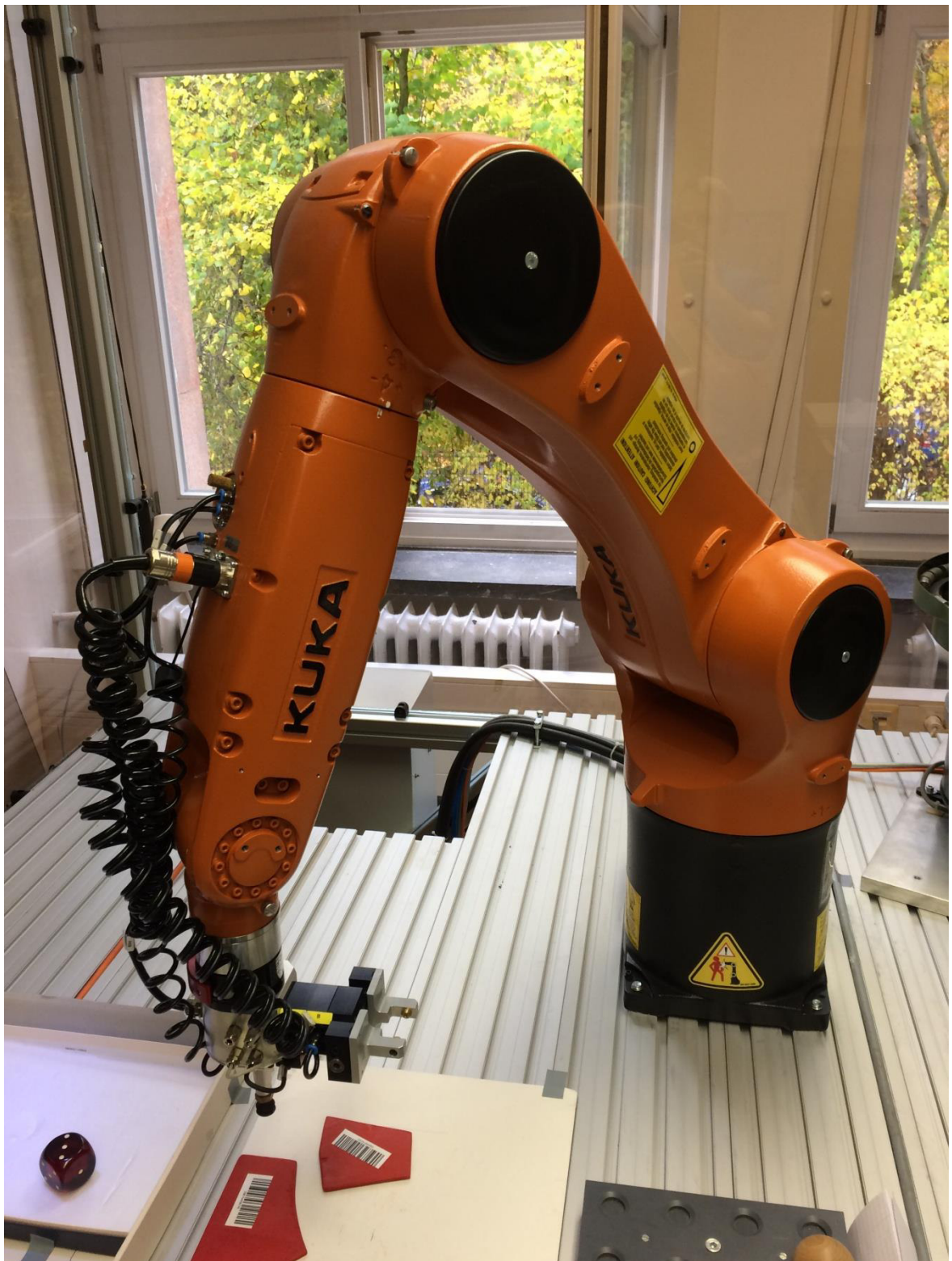


Abbildung 4: Der Roboterarm

- Typ: KUKA AGILUS KR 6 R900 sixx
- Reichweite: 900mm

Die Werkzeuge

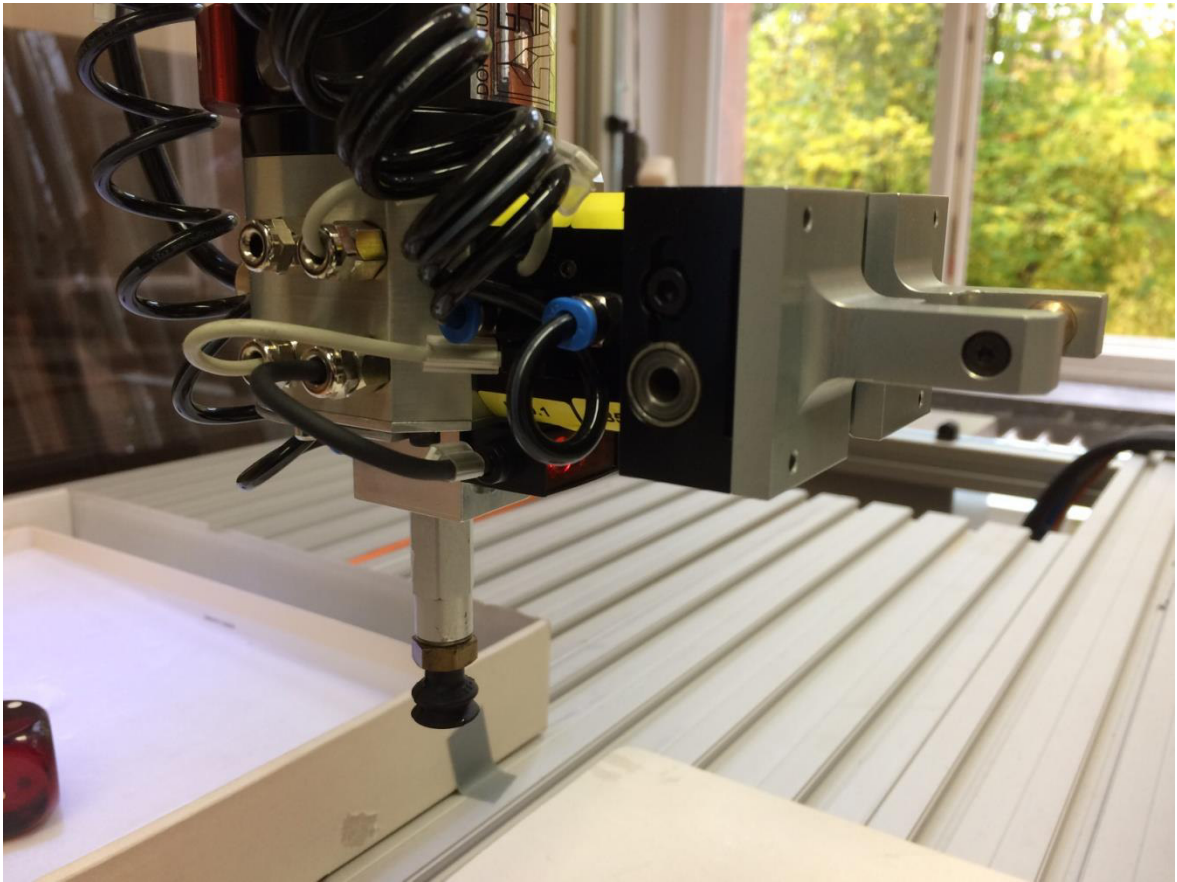


Abbildung 5: Die Werkzeuge

- Sauger senkrecht
- Greifer waagrecht

Die Kamera



Abbildung 6: Die Kamera

- Fabrikat: KUKA Cognex
- Typ: MXG20-17-K40
- Auflösung: 2 MP (1632 x 1235)
- Farben: Schwarz/Weiß
- Bit-Tiefe: 8 Bit

Die Ausleuchtung



Abbildung 7: LED-Ringleuchte

- LED-Ringleuchte mit 24 LEDs (3 sind durch den Umbau kaputt gegangen)

1.2 Die Programme auf dem Steuerungs-PC

- WorkVisual: Zum Übersetzen des Bildes in ein visuelles Objekt

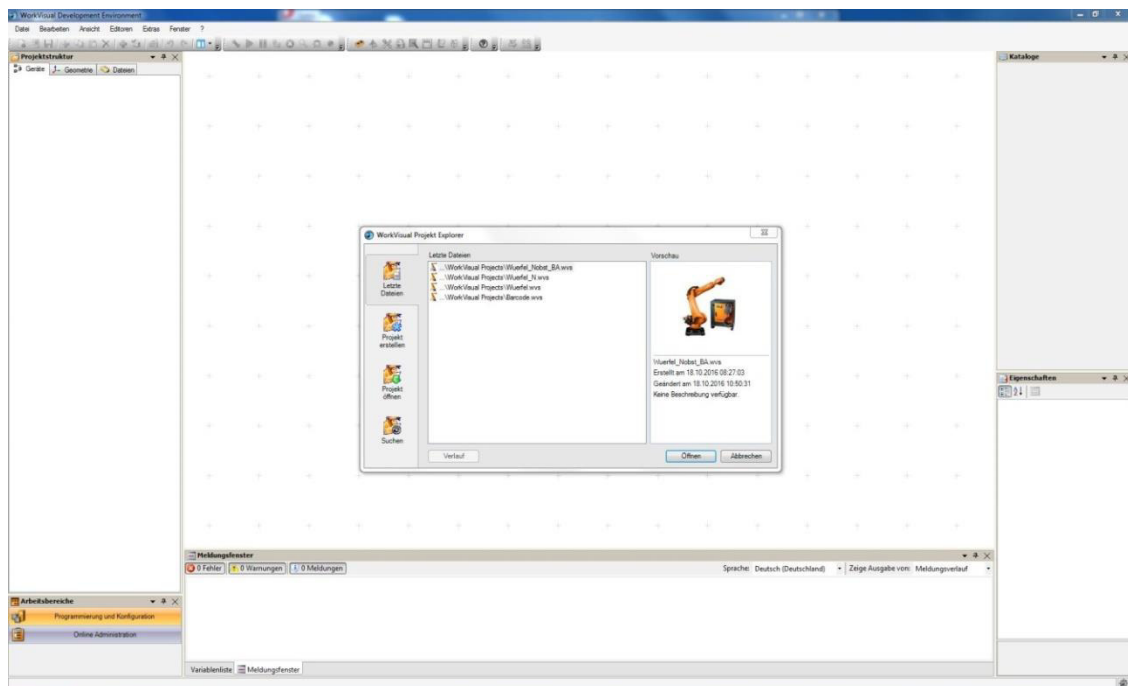


Abbildung 8: WorkVisual-Bedienoberfläche

- TeamViewer: zur Verbindung des PC mit der KR-Steuerung

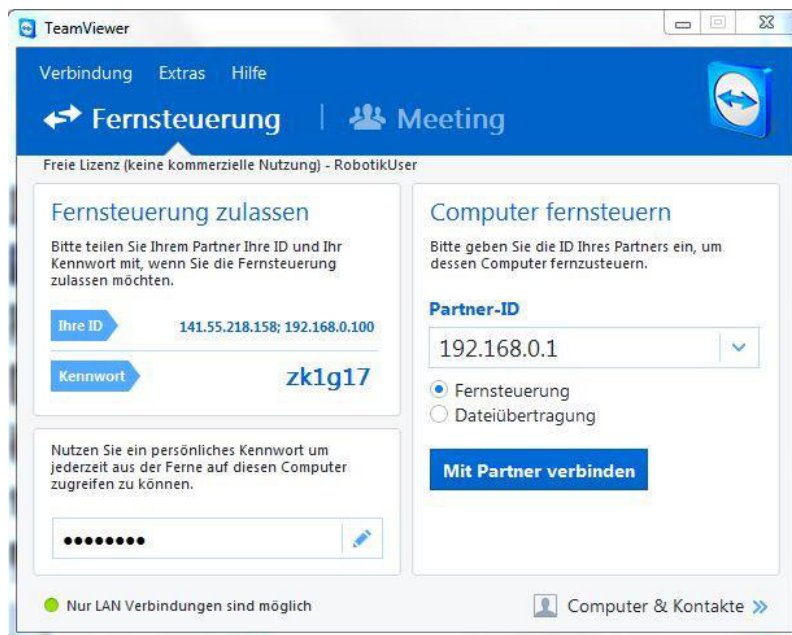


Abbildung 9: TeamViewer-Bedienoberfläche

1.3 Die Programme und das UI auf dem Handheld der KRC

Das UserInterface des KRC-OS wird wie folgt dargestellt. Die Ansicht ist wie abgebildet auf dem Touchpanel zu sehen. Die komplette Oberfläche des Touchpanel kann direkt und ebenso über den PC auf TeamViewer benutzt werden. Nachteil der Nutzung über TeamViewer ist die verlängerte Reaktionszeit. Zahlen, Werte und Texte können über die Tastatur des PC eingegeben werden.

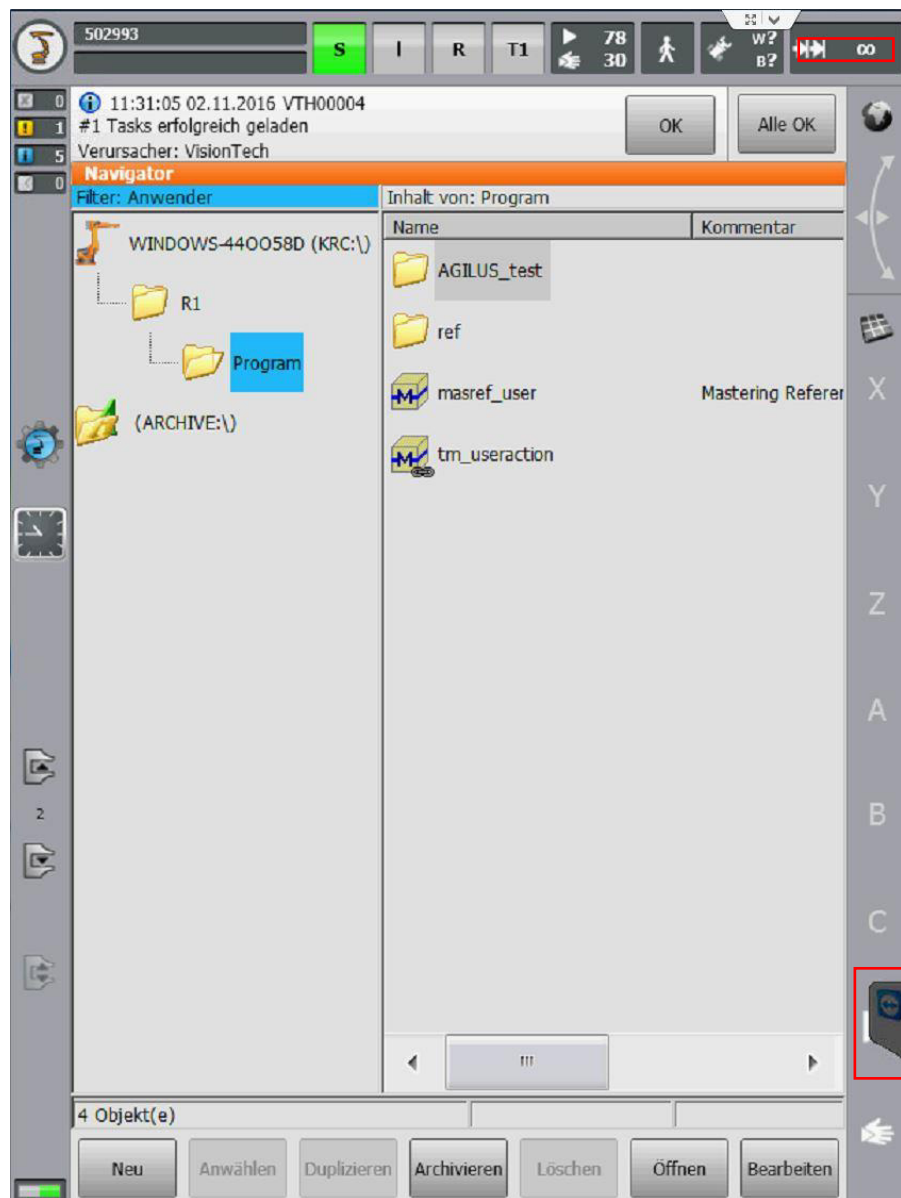


Abbildung 10: Oberfläche des Touchpanels

- TeamViewer wird, wie auf vorherstehender Ansicht als kleine Kästen dargestellt (rote Umrahmung)

- Das Hauptmenü enthält folgende Untermenüs

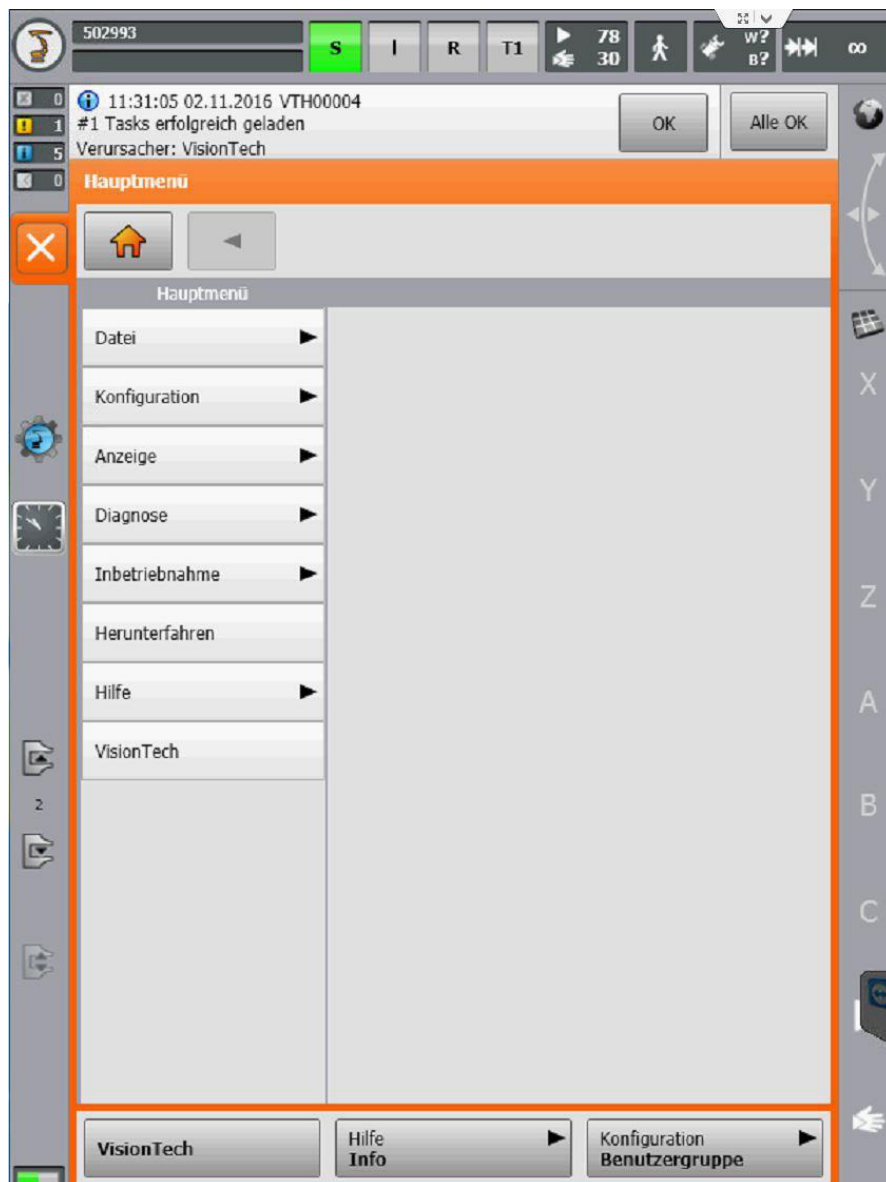


Abbildung 11: Untermenüs im UI

- Das wichtigste aber auch sehr spezielle Programm ist „VisionTech“

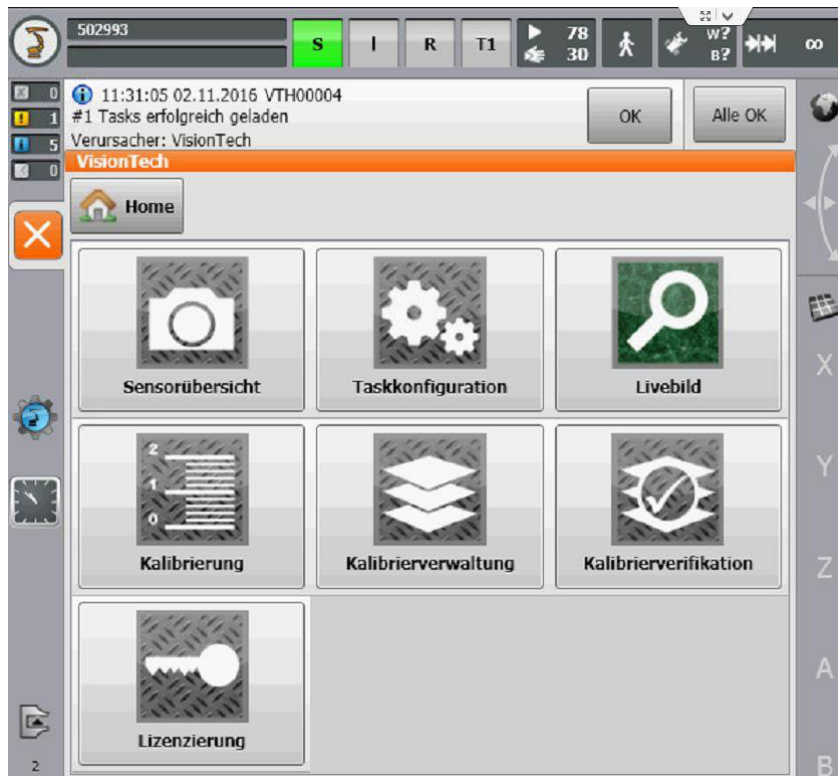


Abbildung 12: VisionTech-Menü

- Die Sensorübersicht beinhaltet alle Sensoren. In unserem Fall jetzt ein optischer Sensor (Die Kamera). Hier können neue Sensoren eingepflegt werden. Dieses Menü wird für diese Aufgabe nicht benötigt.
- Über die Taskkonfiguration können neue Tasks, zum Erstellen eines neuen Modells, angelegt oder vorhandene geändert werden.
- Das Livebild benötigen wir zum Konfigurieren der Helligkeit und der Schärfe der Kamera.
- Die Kalibrierung wird nun zur Kamerakalibrierung benötigt
- Die Kalibrierverwaltung benötigen wir um eine neue Kalibrierebene, auf Basis einer neuen oder bereits vorhandenen Basisvermessung, zu erstellen.
- Nicht benötigt wird für diese Aufgabe die Kalibrierverifikation
- Die Lizenzierung wurde beim Einrichten des Arbeitsplatzes für die Lizenzfreigabe benötigt.

2 Aufgabenstellung

2.1 Ausgangssituation an der Roboterzelle

Die Ausgangssituation hat sich mit der bereits aufgebauten und funktionsfähigen Roboterarbeitszelle dargestellt. Wie in den Voraussetzungen erwähnt, waren die Programme und Funktionen im Vorfeld installiert worden und funktionsfähig. Das zur Bearbeitung notwendige „VisionTech“ war bereits auf der Steuerung installiert und arbeitsfähig. Ebenso war die notwendige Kamera angebaut und funktionsfähig. Die bereits installierte und auch funktionsfähige Version von „WorkVisual 3.1“ auf dem zugehörigen Steuerungs-PC konnte ohne Probleme genutzt werden. Für die optimale Ausleuchtung der Arbeitsebene wurde eine LED-Ringleuchte (Abb. Beleuchtung 2) eingerichtet und angebaut.

2.2 Ziel der Arbeit

Ziel des Bachelorprojektes ist anhand der Kamera das vorgegebene Objekt zu erkennen und dessen Position und Orientierung an die Robotersteuerung zu übergeben. Das zugehörige Programm besteht aus zwei Dateitypen, zum einen die „*.dat“ welche Koordinaten und Werte enthält und dem eigentlichen Programmablauf in der „*.src“. Als sogenanntes Kompatibilitätsprogramm soll im VisionTech ein Task für genannte Würfel eingebunden werden und sorgt für die Kommunikation zwischen der Bildverarbeitung im WorkVisual und der Hardware im VisionTech auf der KRC-Steuerung.

Diese „Source“-Datei sollte als Ziel des Projekts entwickelt werden, um mit dem Greifer die Würfel nacheinander auf vorher programmierte Positionen zu transportieren. Dazu war es notwendig, mehrere Basen einzulernen und zu nutzen. Außerdem wurde eine Matrix für die Würfelablage programmiert.

3 Durchführung

Diese Schritt-für-Schritt-Beschreibung soll es möglich machen, mit vorher besagten Kenntnissen das Einrichten der Kamera und der Kommunikation zwischen den Systemen durchzuführen. Und den Versuch bei Beachtung aller Vorgaben durchführen zu können.

3.1 Einrichten des Bildbereichs

- Als Begrenzung wird eine provisorisch angeordnete „Kiste“ aus fester Pappe genutzt. Diese ist wegen ihrer Maße und Festigkeit gewählt worden.



Abbildung 13: Arbeitsbereich (Kasten)

3.2 Auswahl und Einrichten der Beleuchtung

- Zur Auswahl der Beleuchtung stehen uns verschiedene Arten von Lampen und Leuchtmitteln zur Verfügung. Wir haben uns für eine LED-Ringlampe (*siehe Abb. Beleuchtung 1 links*) entschieden, um die Schattenbildung so gering wie möglich zu halten und die optimale Ausleuchtung zu garantieren.
- Dazu musste der Innenradius der Ringlampe auf die Größe der Abdeckung der Kamera (*siehe Abb. Beleuchtung 1 rechts*) angepasst und montiert werden.
- Zusammengebaut ist diese nun fertig zur Nutzung. (*siehe Abb. Beleuchtung 2*)



Abbildung 14: Beleuchtung 1



Abbildung 15: Beleuchtung 2

3.3 Kalibrieren der Kamera

- Zum Kalibrieren der Kamera wird zuerst eine Kalibrierplatte benötigt, sie legt im Folgenden das interne Maß der Kamera fest. Wir nutzten eine Platte mit den Abmessungen (40x30@3.175mm) wie auf *Abb. Kalibrierplatte* zu sehen ist.

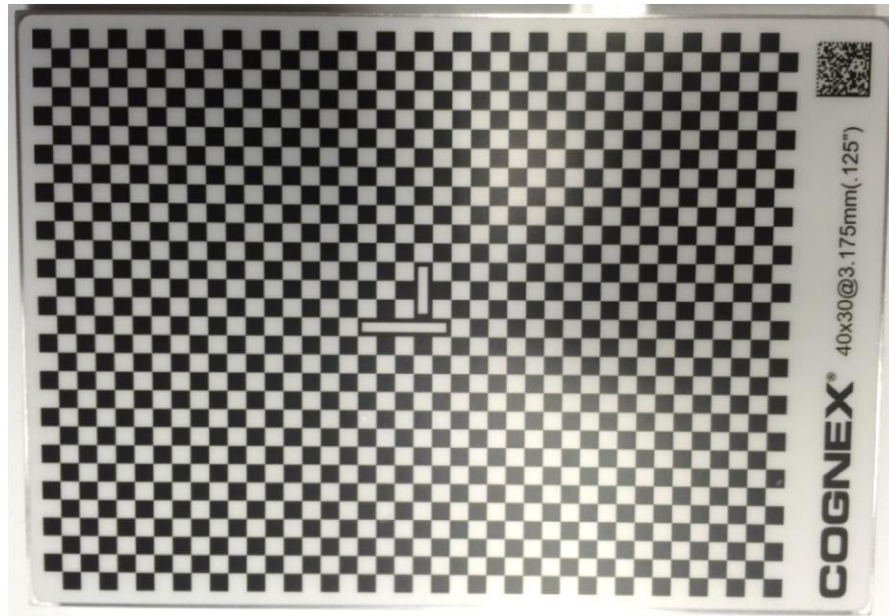



Abbildung 16: Kalibrierplatte

- Diese wird im Sichtbereich der Kamera positioniert, dabei ist auf die COGNEX-Schrift zu achten.
- Um die Orientierung einzuhalten sollte die Schrift bei Draufsicht von vor der Zelle rechts zu finden sein.
- Die sichtbare Oberfläche der Kalibrierplatte sollte nahezu auf Höhe der Sichtkante des Objektes liegen. Notfalls sollten Erhöhungen unter der Platte positioniert werden, um die geforderte Höhe zu erreichen.
- Nun geht man über das in der oberen linken Ecke befindliche -Symbol in das bereits installierte und funktionsfähige Programm **VisionTech** auf dem Touchpanel.
- In diesem Menü die Option **Livebild** wählen, womit wir zur folgenden Ansicht kommen.

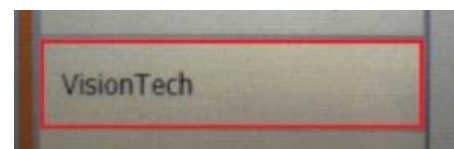


Abbildung 17: Symbol VisionTech

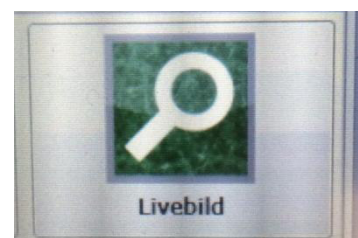


Abbildung 18: Symbol Livebild

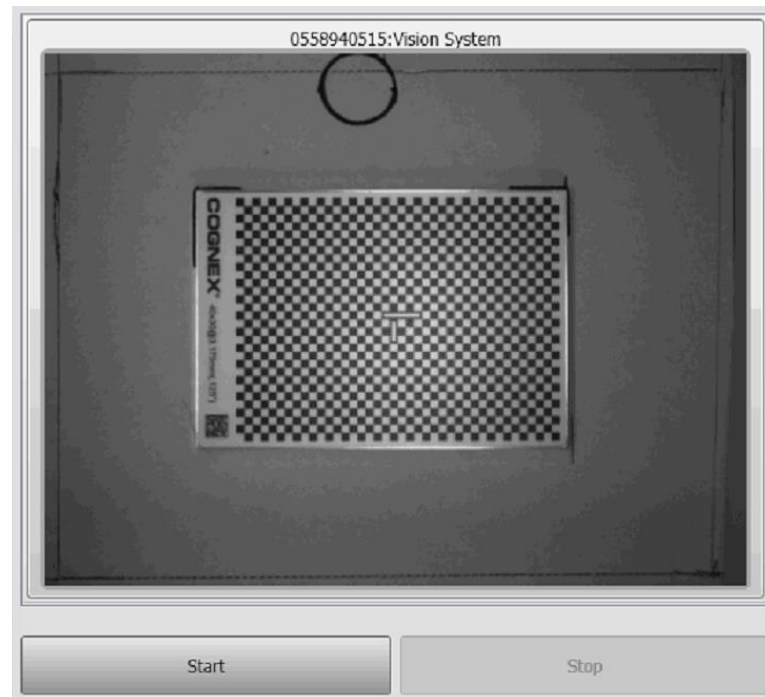


Abbildung 19: Livebild Kalibrierung

- Über die Schaltfläche **Start** das Livebild starten.
- Die Blenden- und Schärfenregelung erfolgt direkt über das Objektiv der Kamera
- Über einen Klick auf das Bild die Belichtungszeit (erfahrungsgemäß die besten Ergebnisse zwischen 40ms) einstellen.
- Nun die Kalibrierbelichtung setzen und über **X** dieses Menü verlassen.
- Im VisionTech-Menü **Kalibrierung** wählen.

Abbildung 20: Button Kalibrierung



- Bereits untergelegten Kalibrierkörper in Slide-Down-Menü wählen.

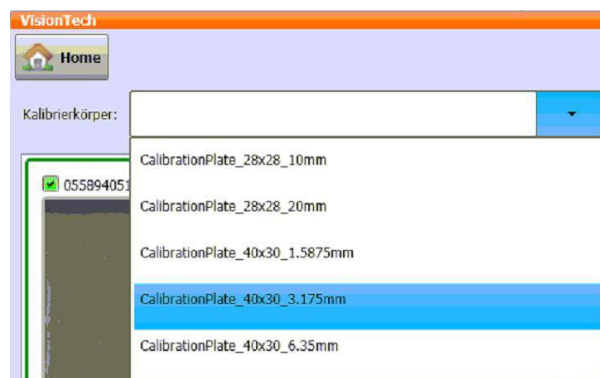


Abbildung 21: Untermenü Kalibrierkörper

- Durch einen Klick **auf das Bild** markieren und den **Kalibrierassistent** auswählen.



Abbildung 22: Auswahl Kalibrierassistent

- Über **Bild aufnehmen** das zusehende Bild speichern und auf **Kalibrieren** klicken.

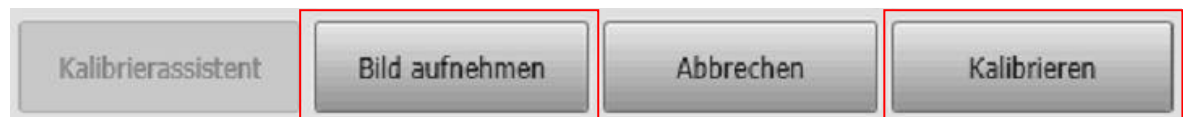


Abbildung 23: Fortsetzung Kalibrierassistent

- Warten bis der kalibriervorgang abgeschlossen ist.

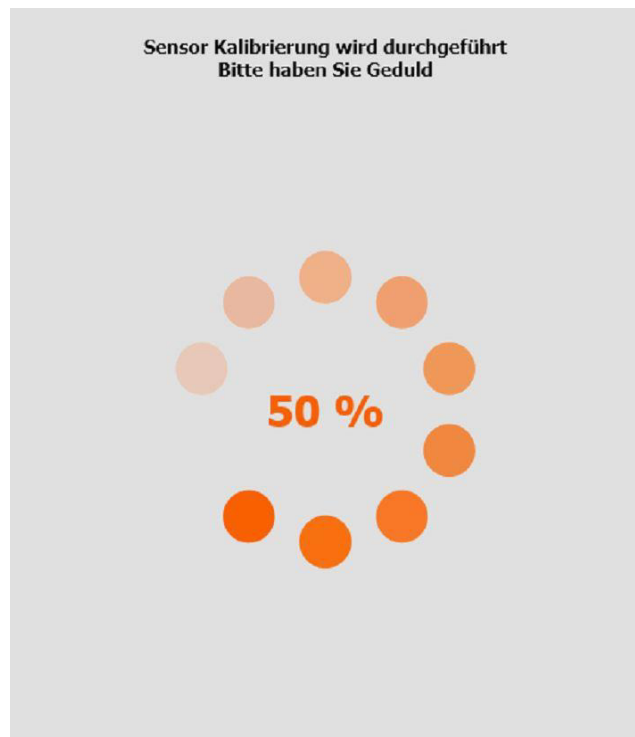


Abbildung 24: Kalibrierungsvorgang

- Eine neue **Kalibrierebene erstellen** und benennen. Die Zeile Beschreibung dient der Wiedererkennung im Falle mehrerer Kalibrierebenen.

Kalibrierebene

Name

Beschreibung

Basis ID

Basis Frame

Neue Kalibrierebene anlegen...

2307_test1

barcodes

EbenefurSLneu

fmkalibase

kalibase_20160119

Abbildung 25: Neue Kalibrierebene

- Im Anschluss die **Basis-ID neu einmessen**.

Mittlerer Fehler [mm]

0.0463635673967196

Kalibrierebene

Name

Beschreibung

Basis ID

Basis Frame

Neue Kalibrierebene anlegen...

TESTNAME

EIGENE BESCHREIBUNG

Basis neu einmessen...

Basis[1]: DESK

Basis[2]: DESK_BIG

Basis[3]: Vision_test1

Basis[4]: ext_WZ

Basis[5]: KalibrierbasisSL





Kalibrierassistent

Bild aufnehmen

Beenden

Kalibrieren

Abbildung 26: neue Basis einmessen

- Eine zu vermessende Basis wählen (nicht bereits benannt) und erkennbar benennen.
- Die neue Basis-ID eingeben.
- Das Werkzeug 3 zum Vermessen wählen (Stift 1 in Greifer einspannen)
- Die auf dem Touchpanel angezeigten Anweisungen befolgen (es ist ratsam, den Ursprung in der Mitte  der Kalibrierplatte zu wählen)
- Dieser ist auf der nächsten Seite zu sehen.
- Die Platte so ausrichten, dass sie wie vorgegeben orientiert zur Kamera liegt.
- Beim Vermessen der positiven Achsen ist darauf zu achten, dass X und Y-Achsen laut Bild ausgerichtet werden müssen (Achsrichtung: , , ) , um mit dem Koordinatensystem der Kamera übereinzustimmen. Die Koordinaten stehen wie folgt: Die positive X-Achse liegt in Y- und die positive Y-Achse in X-.

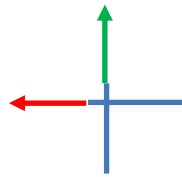



Abbildung 27: Lage des Basiskoordinatensystem



- Nach erfolgreich vermessener Basis über **Speichern** und über  das Menü verlassen.
- Über das **VisionTech**-Menü kommt man direkt zum **Kalibrierassistent**.
- Dort noch einmal **Speichern** und **Beenden**.
- Damit ist die Kalibrierung der Kamera abschlossen.

3.4 Trainieren des zu erkennenden Objekts in WorkVisual

- Im **VisionTech**-Menü (Über **Home**) auf den Punkt **Taskkonfiguration**.



Abbildung 28: Button Taskkonfiguration

- Über **Neu** neuen Task „2D-stationär“ anlegen und erkennbar benennen.
- Herunterscrollen und **New_task** mit linksklick umbenennen.
- Kamerasensor wählen (Haken mit grünem Hindergrund)

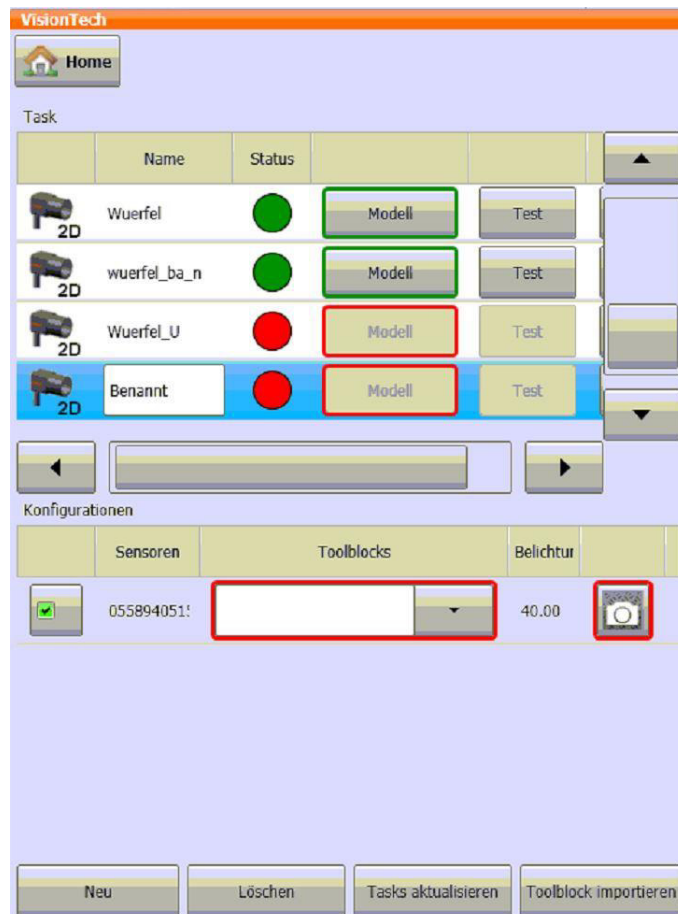




Abbildung 29: Auswahl Kamerasensor

- TCP des Stift_1 über Koordinatenursprung ausrichten.
- Objekt unterlegen und mittig anpassen und so gut wie möglich rechtwinklig anpassen.
- Kalibrierplatte entfernen.
- Roboterarm aus dem Kamerasichtbereich entfernen.
- Über das **Kamerasymbol** ist das nun untergelegte Objekt zu sehen.
- Es kann die Belichtungszeit noch einmal angepasst werden.
- Eingemessene Kalibrierebene auswählen.
- Über **Bild speichern** wird das Bild auf der KRC-Steuerung unter C:\KRC\TP\VisionTech\Snapshots\...und dem selbst gewählten Tasknamen gespeichert.
- Das Bild via USB-Stick auf den PC übertragen.
- Auf dem PC das bereits installierte und voll funktionsfähige WorkVisual starten.
- Über das -Symbol VisionTech öffnen und einen neuen 2D-Toolblock erstellen.
- Über den Rider **Bildquelle** wird das auf dem USB-Stick gespeicherte Objektbild oder der selbstangelegte Sammelordner aller Objektbilder geöffnet.
- Durch Linksklick auf das -Symbol die Toolbox öffnen.
- **Kuka / 2D / LocatePartsOneStage** (Fallouts /) wählen.

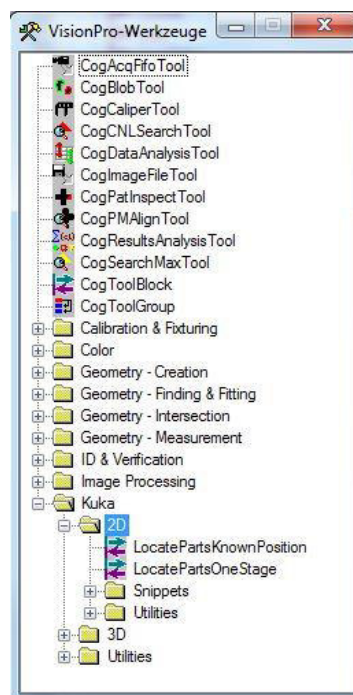


Abbildung 30: VisionPro-Toolbox

- Dieses in die Werkzeugliste zwischen Inputs und Outputs ziehen.
- Wie vorgegeben verbinden:

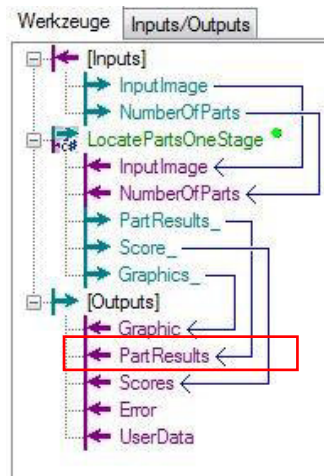



Abbildung 31: Vervollständigen der Werkzeugliste

- Mit dem -Symbol wird das geöffnete Bild angezeigt.
- Über einen Doppelklick auf **LocatePartsOneStage** und **PatMax** das Fenster zum Eintrainieren des Objekts öffnen.
- Durch Rechtsklick auf das Bild und die Option **Bild angleichen** kann das Bild vergrößert werden.
- Das Bild über **Akt. Bild als Ref. Speichern** speichern.
- Im Rider **Grafiken** müssen nun Häkchen wie folgt gesetzt werden:

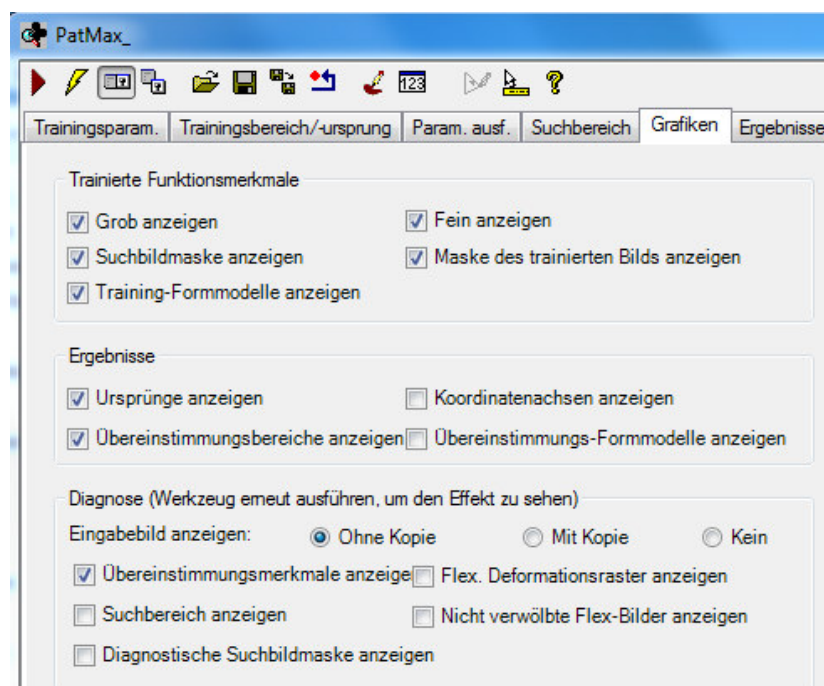


Abbildung 32: PatMax-Menü

- **Grob anzeigen** für grobe Umrisse und Strukturen
- Den linken Rider **Trainingsparam.** wählen und in der SlideDown-Zeile über dem Bild **Current.TrainImage** klicken.
- Den Kasten über die Größe und die Orientierung an das Objekt anpassen.
- Den Ursprung über den Rider **Trainingsbereich/-ursprung** zentrieren.
- Auf **Trainingsparameter** den Button **Trainieren** zum Trainieren klicken.
- Zum Einstellen der Laufzeitparameter den Menüpunkt **Param. ausf.** wählen und die Werte laut dem folgenden Bild einstellen.
- Der Maßstab sollte auf 0,9 und 1,1 gestellt werden um Verzerrungen und optische Größenunterschiede umgehen zu können.


Zone	Nominal	Niedrig	Hoch
Winkel	0 deg	-45 deg	45 deg
Maßst.	1	0,9	1,1

Abbildung 33: Parameterausführung

- Im Falle einer Kugel oder eines für die Kamera runden Objektes sollte der Winkel über den Pfeil auf „0“ gesetzt werden.

Zone	Nominal	Niedrig	Hoch
Winkel	0 deg	-45 deg	45 deg

Abbildung 34: Winkeleinstellung

- Das PatMax- und das LocatePartsOneStage-Fenster schließen.
- Wieder auf das -Symbol klicken, um die Einstellungen zu aktualisieren.
- Durch einen Klick auf das Speichern-Symbol und die Auswahl „**Vollständiges Werkzeug speichern...**“ den Toolblock auf dem USB-Stick speichern.
- Auf die KRC-Steuerung unter C:\KRC\USER\ProjectRoot\...und einen selbst erstellten Ordner kopieren.

- Im **VisionTech**-Menü den Punkt **TaskKonfiguration** wählen.
- Für den erstellten Task den Toolblock aus dem angelegten Ordner importieren



Abbildung 35: Ordnerstruktur für Taskkonfiguration

- Für den Sensor auswählen.

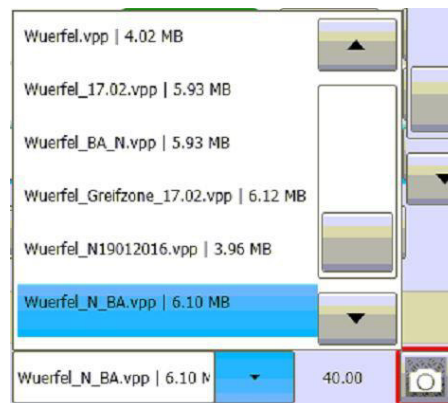


Abbildung 36: Sensorauswahl

- Das einzulernende Objekt *lt. 3.5 Definieren der Nulllage* des Objektes auf den Nullpunkt des Kamerasystems (Nullpunkt der Kalibrierplatte) legen.
- Über den rot umrahmten Button das Objekt öffnen und speichern.
- Das in Referenzposition liegende Teil als Modell erzeugen.
- Dieses wird nun als Objekt erkannt und wird mit zugehörigen Koordinaten angezeigt werden.

3.5 Definieren der Null-Lage des Objekts

Die Null-Lage sollte nach Möglichkeit in der Mitte ggf. im Kamerasystem-Nullpunkt liegen. Da dieser mit der Kalibrierplatte erstellt wurde, ist es ratsam, diese als Anhaltspunkt zur Positionierung zu nutzen.

- Kalibrierplatte über Markierung ausrichten.
- Mit Hilfe der Stiftspitze des in den Roboter eingespannten Stift 1 auf den Nullpunkt zeigen.

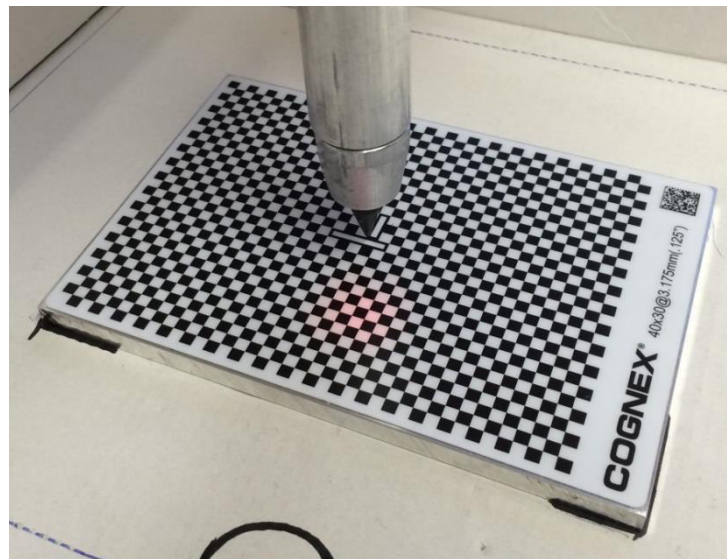


Abbildung 37: Anfahren des Nullpunkts

- Den Stift etwas anheben, um das Objekt positionieren zu können.

3.6 Einlernen (teachen) der Zeigerlage

- Die Zeigerlage wird nach ausgeführter Zeile 36 des Programmes ausgeführt.
- Die Punkte müssen in Basis 16 programmiert (geteacht) werden, weil diese die Berechnung für den genauen Ort des Objektes enthält.
- Basis 16 wird aus der Kalibrierbasis und dem CorrFrame zusammengesetzt und vektoriell berechnet.
- Der CorrFrame ist der Vektor der Verschiebung des Objektes zum eigentlichen Base-Nullpunkt.

3.7 Das zugehörige Beispiel-Programm

- Zur Erklärung der verschiedenen Befehle liegt die Datei: KST-VisionTech_30_de.pdf bei. Hier sind die Seiten 73-85 zu beachten.
- Vor dem Schreiben des Programmes prüfen, ob ein Flag für den Interrupt gesetzt ist.
- C:\KRC\ROBOTER\Config\USER\Common\EthernetKRL\VisionTechConfig.xml öffnen und Zeile 37 <....Set Flag=Choose_Your_Flag /> überprüfen und ggf. das Flag 998 setzen

```
1 DEF Muerfel_N_BA( )
2
3 DECL INT Part
4 DECL FRAME CORRECTIONFRAME
5 DECL FRAME CorrFrame
6
7 SET GRP 2 State=BLOW CONT at START Delay=0 ms
8 INI
9
10 $FLAG[1]=FALSE
11 RESULTCOUNTER=0
12
13 INTERRUPT DECL 98 WHEN $FLAG[998]==TRUE DO
  ↳ UT_GETTASKRESULTS("VisionTechConfig",RESULTS[],
  ↳ RESULTCOUNTER,1,998)
14 INTERRUPT ON 98
```

Abbildung 38: Beispielprogramm Teil 1

```
15
16
17
18 ;HIER BITTE EINEN PUNKT IN EINER FESTEN BASIS
   ↳ TEACHEN Z.B. $Nullframe
19 PTP Start Vel=100 % PDAT11 Tool[2]:UAC Base[0]
20 LIN vorpos Vel=2 m/s CPDAT1 Tool[2]:UAC Base[0]
21
22
23 UT_INIT("VisionTechConfig")
24 UT_OPENCONNECTION("VisionTechConfig")
25 UT_TASKTRIGGER_REFBASE("VisionTechConfig",
   ↳ "wuerfel_ba_n",BASE_DATA[8])
26
27 UT_WAITFORRESULT(1)
28
29 FOR PART = 1 TO RESULTCOUNTER STEP 1
30
31 IF (UT_CHECKRESULT(RESULTS[PART])) THEN
32
33 ;HIER WIRD DIE DAS EINZELNE ERGEBNIS AUSGEWERTET
34 CorrFrame=UT_GETCORRECTIONFRAME(RESULTS[PART])
35
36
37 ;HIER DIE BASIS 16 aus Basis 10 berechnet!
38 Base_Data[16]=BASE_DATA[10]:CorrFrame
39 HALT
40 ;HIER PUNKTE IN BASIS 16 TEACHEN
41
42 LIN uberw Vel=2 m/s CPDAT7 Tool[2]:UAC Base[16]
```

Abbildung 39: Beispielprogramm Teil 2

```
43 LIN uberw1 Vel=2 m/s CPDAT9 Tool[2]:VAC Base[16]
44 LIN gripw Vel=2 m/s CPDAT3 Tool[2]:VAC Base[16]
45 SET GRP 2 State=VACUUM GDAT3
46 LIN uberw Vel=2 m/s CPDAT4 Tool[2]:VAC Base[16]
47 ELSE
48
49 ;ES WURDE KEIN BAUTEIL GEFUNDEN!
50 HALT
51 ENDIF
52 PTP vorablage Vel=100 % PDAT14 Tool[2]:VAC Base[0]
53 LIN ablage Vel=2 m/s CPDAT10 Tool[2]:VAC Base[0]
54 SET GRP 2 State=BLOW GDAT4
55 LIN vorablage Vel=2 m/s CPDAT11 Tool[2]:VAC Base[0]
56 LIN ablage Vel=2 m/s CPDAT12 Tool[2]:VAC Base[0]
57 SET GRP 2 State=VACUUM GDAT5
58 PTP vorablage Vel=100 % PDAT15 Tool[2]:VAC Base[0]
59 PTP abwurf Vel=100 % PDAT16 Tool[2]:VAC Base[0]
60 SET GRP 2 State=BLOW GDAT6
61 PTP vorablage Vel=100 % PDAT17 Tool[2]:VAC Base[0]
62 ENDFOR
63
64 LIN Start Vel=2 m/s CPDAT5 Tool[2]:VAC Base[0]
65 UT_CloseConnection("VisionTechConfig")
66 UT_Clear("VisionTechConfig")
67
68 END
```

Abbildung 40: Beispielprogramm Teil 3

- Die Punkte über dem Objekt werden erst „geteacht“, wenn Zeile 38 ausgeführt wurde.
- Zeile 38 stellt die Verbindung der festen geteachten Basis zur variablen Basis, welche sich nach Lage des Objekts verschiebt, her. Und synchronisiert diese um die genaue Lage des Objekts an die Steuerung und somit den Roboterarm übergeben zu können.
- Dieser Befehl ist für das Teachen der Objektbezogenen Koordinaten und die damit zusammenhängenden Bewegungen unumgänglich.
- Zur Erklärung der einzelnen VT_*-Funktionen liegen Auszüge aus der im Ordner „VisionTexte“

4 Hinweise auf mögliche Fehlerquellen

4.1 Einrichten der Arbeitsumgebung

- Korrekte Orientierung der Kalibrierplatte
- Korrekte Orientierung der positiven Achsen beim Kalibriervorgang
- Zu große Schattenbildung
- Spiegelung/Reflexion auf dem Objekt
- Uebener Untergrund
- Falsche Blenden- oder Schärfeeinstellung an dem Objektiv der Kamera

4.2 Beim Einlernen des Objektes im WorkVisual

- Reihenfolge des Ablaufs nicht beachtet
- Werte vertauscht oder falsch eingegeben

4.3 Beim Schreiben des Programmes

- Zeile vergessen
- Doppeldeklaration (Eine Funktion wurde doppelt deklariert.)
- Nicht deklarierte Aktionen und Befehle
- Task falsch benannt
- Zuordnungen vertauscht

5 Fazit

Bei meinem Bachelorprojekt habe ich grundlegende Fähigkeiten zur Programmierung, das Verhalten des Roboters und der Funktionalität des Kamerasystems erlangen können. Die Zusammenarbeit mit KUKA war hier nicht nötig, da die Voraussetzungen bereits erreicht waren. Nach einigen Versuchen, das Beispielprogramm funktionsfähig zu machen, habe ich letztendlich eine bereits vorhandene Programmstruktur, welche ich im Laufe meines Praktikums erarbeitet habe, genutzt. Womit sich das Problem nach einiger Arbeit und Umprogrammierung an diesem Programm löste und das Teil nun erkannt und korrekt angefahren und angesaugt wurde. Nun mussten nur noch die Bewegungen und einzelnen anzufahrenden Punkte optimiert und auf das richtige Werkzeug, dem Vakuum-Sauger, eingestellt werden musste.

Literatur und Hilfen

- [Q1] KUKA Roboter GmbH: Training Roboterprogrammierung 1, Augsburg, 2014
- [Q2] KUKA Roboter GmbH: Training Roboterprogrammierung 2, Augsburg, 2014
- [Q3] KUKA Roboter GmbH: VisionTech_de.mp4, Augsburg, 2014
- [Q4] KUKA Roboter GmbH: KST_VisionTech_30_de.pdf, Augsburg, 2015
- [Q5] Prof. Dr. Klaus Müller: Vorlesung Robotik II, Mittweida 2015
- [Q6] Maik Nobst: „Visionsystem für eine Roboteranwendung (KUKA), Praktikumsbericht, Mittweida 2016

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 16.01.2017

Maik Nobst